Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №4**

З дисципліни «Системне програмування»

Виконав: Перевірив:

Студент групи ІО-32 ст. викладач

Попенко Р.Л. Порєв В. М.

Київ 2015

**Мета:** Навчитися програмувати на асемблері основні арифметичні операції

підвищеної розрядності, а також отримати перші навички програмування

власних процедур у модульному проекті.

**Завдання:**

1. Створити у середовищі MS Visual Studio проект з ім’ям **Lab4**.

2. Написати вихідний текст програми згідно варіанту завдання. У проекті

мають бути три модуля на асемблері:

- головний модуль: файл **main4.asm**. Цей модуль створити та написати

заново, частково використавши текст модуля main3.asm попередньої

роботи №3;

- другий модуль: використати **module** попередньої роботи №3;

- третій модуль: створити новий з ім'ям **longop**.

3. У цьому проекті кожний модуль може окремо компілюватися.

4. Скомпілювати вихідний текст і отримати виконуємий файл програми.

5. Перевірити роботу програми. Налагодити програму.

6. Отримати результати – кодовані значення чисел згідно варіанту завдання.

7. Проаналізувати та прокоментувати результати, вихідний текст та

дизасембльований машинний код програми.

Текст програми:

.586

.model flat, stdcall

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

include module.inc

include longop.inc

.data

Caption1 db "A+B",0

Caption2 db "A2+B",0

Caption3 db "A2-B",0

Text db 832 dup(0)

Text2 db 288 dup(0)

ValueA db 832 dup(0)

ValueA2 db 832 dup(0)

ValueB db 832 dup(0)

SubA db 288 dup(0)

SubB db 288 dup(0)

SubResult db 288 dup(0)

Result db 832 dup(0)

.code

main:

mov eax, 80010001h

mov ecx, 0

setValueA:

mov dword ptr [ValueA + ecx], eax

add eax, 00010000h

add ecx, 4

cmp ecx, 832

jl setValueA

mov eax, 80000001h

mov ecx, 0

setValueB:

mov dword ptr [ValueB + ecx], eax

add ecx, 4

cmp ecx, 832

jl setValueB

push offset ValueA

push offset ValueB

push offset Result

call Add\_Long

push offset Text

push offset Result

push 832

call StrHex\_MY

invoke MessageBoxA, 0, ADDR Text, ADDR Caption1, 0

mov eax, 00000018h

mov ecx, 0

setValueA2:

mov dword ptr [ValueA2 + ecx], eax

add eax, 00000001h

add ecx, 4

cmp ecx, 832

jl setValueA2

push offset ValueA2

push offset ValueB

push offset Result

call Add\_Long

push offset Text

push offset Result

push 832

call StrHex\_MY

invoke MessageBoxA, 0, ADDR Text, ADDR Caption2, 0

mov eax, 00000018h

mov ecx, 0

setSubB:

mov dword ptr [SubB + ecx], eax

add eax, 00000001h

add ecx, 4

cmp ecx, 288

jl setSubB

push offset SubA

push offset SubB

push offset SubResult

call Sub\_Long

push offset Text2

push offset SubResult

push 288

call StrHex\_MY

invoke MessageBoxA, 0, ADDR Text2, ADDR Caption3, 0

invoke ExitProcess, 0

end main

Longop.asm

.586

.model flat, c

.code

Add\_Long proc

push ebp

mov ebp,esp

mov esi, [ebp+16]

mov ebx, [ebp+12]

mov edi, [ebp+8]

mov ecx, 0

addAB:

mov eax, dword ptr[esi+ecx]

adc eax, dword ptr[ebx+ecx]

mov dword ptr [edi+ecx], eax

add ecx, 4

cmp ecx, 416

jl addAB

pop ebp

ret 12

Add\_Long endp

Sub\_Long proc

push ebp

mov ebp,esp

mov esi, [ebp+16]

mov ebx, [ebp+12]

mov edi, [ebp+8]

mov ecx, 0

subAB:

mov eax, dword ptr[esi+ecx]

sbb eax, dword ptr[ebx+ecx]

mov dword ptr [edi+ecx], eax

add ecx, 4

cmp ecx, 704

jl subAB

pop ebp

ret 12

Sub\_Long endp

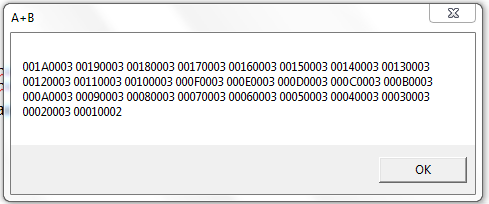
End

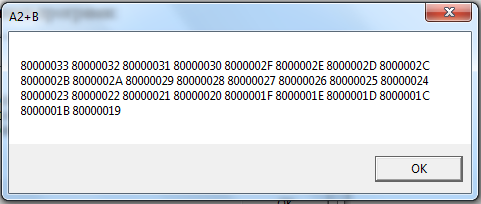
Longop.inc

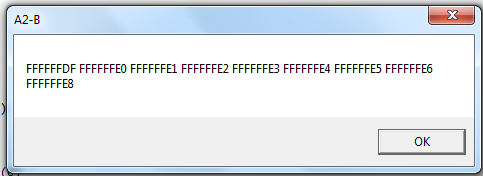
EXTERN Add\_Long : proc

EXTERN Sub\_Long : proc

Результат виконання програми:







Аналіз результатів:

Дана програма виконує операції віднімання і додавання з числами підвищеної точності.

Висновок:

Під час виконання лабораторної роботи були покращені навички написання власних модулів, а також були закріпленні основні навички в операціях додавання і віднімання чисел з підвищеною точністю.